

13. Яцишин А.В. Розвиток інформаційно-дослідницької компетентності молодих вчених у сучасному інформаційному просторі / Професійний розвиток фахівців у системі освіти дорослих: історія, теорія, технології: зб. матеріал. III-ої Всеукр. Інтернет-конф. (18 квітня 2018 р.). – К.: Агроосвіта, 2018. – С. 204-205.

**Кучер В.В.**

*студент 6 курсу*

*фізико-математичного факультету*

*Науковий керівник: **Кривонос О. М.,***

*канд. пед. наук, доцент,*

*доцент кафедри прикладної математики та інформатики*

*Житомирський державний університет імені Івана Франка*

## **МІКРОСЕРВІСНА АРХІТЕКТУРА ТА ЇЇ ОСОБЛИВОСТІ**

Мікросервісна архітектура — варіант сервіс-орієнтованої архітектури програмного забезпечення, орієнтований на взаємодію настільки це можливо невеликих, слабо пов'язаних і легко змінюваних модулів — мікросервісів, що одержав поширення в середині 2010-х років у зв'язку з розвитком практик гнучкої розробки та DevOps[1].

Якщо в традиційних варіантах сервіс-орієнтованої архітектури модулі можуть бути самі по собі досить складними програмними системами, а взаємодія між ними часто покладається на стандартизовані великовагові протоколи (такі, як SOAP, XML-RPC), в мікросервісній архітектурі системи вибудовуються з компонентів, які виконують відносно елементарні функції, і які взаємодіють з використанням економічних мережевих комунікаційних протоколів (в стилі REST з використанням, наприклад, JSON, Protocol Buffers, Thrift). За рахунок підвищення гранулярності модулів архітектура націлена на посилення ступеня

зацеплення і зменшення пов'язаності, що дозволяє простіше додавати і змінювати функції в системі в будь-який час.

Незважаючи на те, що термін «мікросервіси» згадувався з середини 2000-х років, появу терміну прийнято вважати 2011 рік. У 2012 році про мікросервіси доповідали на конференції 33d Degree в Кракові. У 2012-2014 роках про впровадження мікросервісів у рамках власних програмних розробок заявляли фахівці таких компаній як Amazon, Netflix, Twitter, з 2015 року у провідних видавництвах регулярно випускається спеціалізована література, систематично проводяться тематичні конференції, присвячені мікросервісам.

Опишемо основні властивості, що характерні для мікросервісної архітектури:

- ◆ модулі можна легко замінити в будь-який час: акцент на простоту, незалежність розгортання та оновлення кожного з мікросервісів;
- ◆ модулі організовані навколо функцій: мікросервіс по можливості виконує тільки одну досить елементарну функцію;
- ◆ модулі можуть бути реалізовані з використанням різних мов програмування, фреймворків, сполучного програмного забезпечення, виконуватися в різних середовищах контейнеризації, віртуалізації, під управлінням різних операційних систем на різних апаратних платформах: пріоритет віддається на користь найбільшої ефективності для кожної конкретної функції перед стандартизацією засобів розробки і виконання;
- ◆ архітектура симетрична, а не ієрархічна: залежності між мікросервісами однорангові.[2]

Ідеологія мікросервісів фактично копіює концепцію Unix, згідно з якою кожна програма повинна «робити щось одне, і робити це добре» та взаємодіяти з іншими програмами простими засобами: мікросервіси мінімальні та призначаються для єдиної функції. Основні зміни у зв'язку з цим накладаються на організаційну взаємодію, яка повинна включати автоматизацію розробки та тестування, а також культуру проектування, від

якої потрібно передбачати обхід колишніх помилок, виключення по можливості успадкованого коду (мікросервіси часто замінюють цілком, оскільки їх функції елементарні).

Найпопулярніше середовище для виконання мікросервісів є системи управління контейнеризованими додатками (такі як Kubernetes і її надбудови OpenShift і CloudFoundry[en], Docker Swarm, Apache Mesos[en]), в цьому випадку кожен з мікросервісів як правило ізолюється в окремий контейнер або невелику групу контейнерів, доступну по мережі іншим мікросервісам і зовнішнім споживачам, і управляється середовищем оркестризації, що забезпечує відмовостійкість і балансування навантаження.

Архітектура постійно піддається критиці з самого моменту її формування, серед нових проблем, які виникають при її впровадженні відзначаються:

- ◆ мережеві затримки: якщо в модулях, що виконують кілька функцій, взаємодія локально, то мікросервісна Архітектура накладає вимогу атомізації модулів і взаємодії їх по мережі;

- ◆ формати повідомлень: відсутність стандартизації та необхідність узгодження форматів обміну фактично для кожної пари взаємодіючих мікросервісів призводить як до потенційних помилок, так і складнощів налагодження;

- ◆ баланс навантаження і відмовостійкості.

- ◆ На додаток до цього, ускладнюється тестування і розробка.

Отже ми проаналізували основні переваги та недоліки технології мікросервісної архітектури. Саме на етапі проектування програмного продукту розробникам варто зважити всі плюси, мінуси та прийняти рішення про необхідність мікросервісного або звичаного монолітного підходу.

#### **Список використаних джерел:**

1. Мікросервіси [Електронний ресурс] – Режим доступу:

<https://uk.wikipedia.org/wiki/Мікросервіси> - Назва з екрана.

2. Микросервисы (Microservices) [Електронний ресурс] – Режим доступу: <https://habr.com/post/249183/> - Назва з екрана.

3. Преимущества и недостатки микросервисной архитектуры [Електронний ресурс] – Режим доступу: <https://eas.me/micro-service-architecture/> - Назва з екрана.

**Семенюк Р. А.**

*студент 6 курсу*

*фізико-математичного факультету*

*Науковий керівник: **Кривонос О. М.,***

*канд. пед. наук, доцент,*

*доцент кафедри прикладної математики та інформатики*

*Житомирський державний університет імені Івана Франка*

### **MICROSOFT DYNAMICS NAV**

Microsoft Dynamics NAV[1] — система планування ресурсів підприємства (ERP), що була розроблена датською компанією Navision. Наразі є одним з програмних продуктів компанії Microsoft і входить до лінійки Microsoft Dynamics.

Система є трирівневим клієнт-серверним додатком, що складається з системи управління базами даних, сервера додатка, а також користувачів, що підключаються за допомогою Windows або Веб-інтерфейсу.

В якості системи управління базою даних використовується Microsoft SQL Server. Аналітичні служби та служби звітності SQL Server Використовуються для аналізу і обробки операційних даних. Адаптованість Microsoft Dynamics NAV забезпечується об'єктно-орієнтованої середовищем розробки C / SIDE (Client / Server Integrated Development Environment). C / SIDE має відкритий вихідний код, що дозволяє вільно змінювати код програми: допрацьовувати стандартну функціональність, створювати індивідуальну звітність і т. д.